

## Programming Assignment # 4 Jacob Sunia

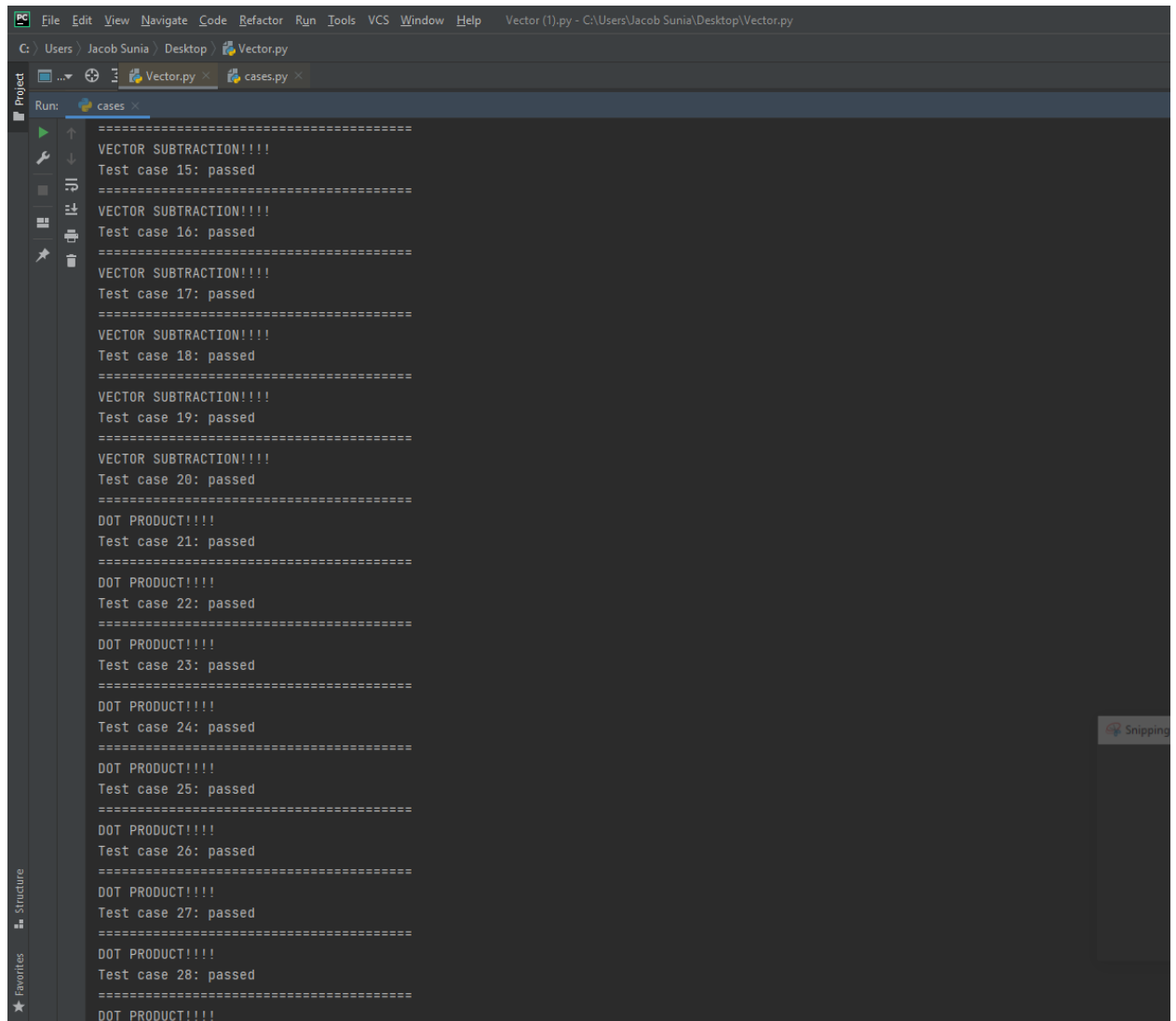
Code:

```
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help Vector (1).py - C:\Users\Jacob Sunia\Desktop\Vector.py
C:\Users\Jacob Sunia\Desktop\Vector.py
Vector.py x cases.py x
main.py 1 """
2     Name: Jacob Sunia
3     Time Finished: 0859 20 April 2021
4     Date Due: 23 April 2021
5 """
6 class Vector:
7
8     # input: l - a list
9     # Example: Vector([1, 2, 3, 4]) will make a vector <1, 2, 3, 4>
10    def __init__(self, l):
11        # ALREADY DONE FOR YOU! DO NOT TOUCH
12        self.v = l
13        self.size = len(l)
14
15    # get's the ith element of the vector
16    def get_ith_element(self, i):
17        return self.v[i]
18
19    # Part0
20    # vector addition (ALREADY IMPLEMENTED)
21    # inputs: self, other
22    # output: if addition is possible, return the sum
23    #         if addition is not possible, return None
24    def __add__(self, other):
25        if self.size != other.size:
26            return None
27        sum = []
28        for i in range(self.size):
29            sum.append(self.get_ith_element(i) + other.get_ith_element(i))
30        return Vector(sum)
31
32    # Part1
33    # TODO: implement vector subtraction
34    # inputs: self, other
35    # output: if subtraction is possible, return the differences
36    #         if subtraction is not possible, return None
37    def __sub__(self, other):
38        if self.size != other.size:
39            return None
40        diff = []
41        for i in range(self.size):
42            diff.append(self.get_ith_element(i) - other.get_ith_element(i))
43        return Vector(diff)
```

```
PC File Edit View Navigate Code Refactor Run Tools VCS Window Help Vector (1).py - C:\Users\Jacob Sunia\Desktop\Vector.py
C:\Users\Jacob Sunia\Desktop\Vector.py
Vector.py cases.py
main.py
External I
Scratches
24 def __add__(self, other):
25     if self.size != other.size:
26         return None
27     sum = []
28     for i in range(self.size):
29         sum.append(self.get_ith_element(i) + other.get_ith_element(i))
30     return Vector(sum)
31
32 # Part1
33 # TODO: implement vector subtraction
34 # inputs: self, other
35 # output: if subtraction is possible, return the differences
36 #         if subtraction is not possible, return None
37 def __sub__(self, other):
38     if self.size != other.size:
39         return None
40     diff = []
41     for i in range(self.size):
42         diff.append(self.get_ith_element(i) - other.get_ith_element(i))
43     return Vector(diff)
44
45 # Part2
46 # TODO: implement dot product
47 # inputs: self, other
48 # output: if dot product is possible, return the dot product
49 #         if dot product is not possible, return None
50 def __mul__(self, other):
51     if self.size != other.size:
52         return None
53     mul = []
54     for i in range(self.size):
55         mul.append(self.get_ith_element(i) * other.get_ith_element(i))
56     return sum(mul)
57
58 # DO NOT TOUCH! For debugging purposes
59 def __str__(self):
60     return str(self.v)
61
```

Output:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Vector (1).py - C:\Users\Jacob Sunia\Desktop\Vector.py
C:\Users\Jacob Sunia\Desktop\Vector.py
Vector.py cases.py
Run: cases
"C:\Users\Jacob Sunia\AppData\Local\Microsoft\WindowsApps\python3.9.exe" "C:/Users/Jacob Sunia/Desktop/cases.py"
VECTOR ADDITION!!!!
Test case 1: passed
=====
VECTOR ADDITION!!!!
Test case 2: passed
=====
VECTOR ADDITION!!!!
Test case 3: passed
=====
VECTOR ADDITION!!!!
Test case 4: passed
=====
VECTOR ADDITION!!!!
Test case 5: passed
=====
VECTOR ADDITION!!!!
Test case 6: passed
=====
VECTOR ADDITION!!!!
Test case 7: passed
=====
VECTOR ADDITION!!!!
Test case 8: passed
=====
VECTOR ADDITION!!!!
Test case 9: passed
=====
VECTOR ADDITION!!!!
Test case 10: passed
=====
VECTOR SUBTRACTION!!!!
Test case 11: passed
=====
VECTOR SUBTRACTION!!!!
Test case 12: passed
=====
VECTOR SUBTRACTION!!!!
Test case 13: passed
=====
VECTOR SUBTRACTION!!!!
Test case 14: passed
=====
VECTOR SUBTRACTION!!!!
```



The screenshot shows an IDE window with the following components:

- Menu Bar:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help.
- Toolbar:** Includes icons for file operations (new, open, save, close), search, and running code.
- Project Explorer:** Located on the left, showing a project named 'Vector.py' with a sub-file 'cases.py'.
- Run Console:** Displays the output of the executed script. The output consists of a series of test cases for vector operations, each preceded by a separator line of equals signs.

The output in the Run Console is as follows:

```
=====
VECTOR SUBTRACTION!!!!
Test case 15: passed
=====
VECTOR SUBTRACTION!!!!
Test case 16: passed
=====
VECTOR SUBTRACTION!!!!
Test case 17: passed
=====
VECTOR SUBTRACTION!!!!
Test case 18: passed
=====
VECTOR SUBTRACTION!!!!
Test case 19: passed
=====
VECTOR SUBTRACTION!!!!
Test case 20: passed
=====
DOT PRODUCT!!!!
Test case 21: passed
=====
DOT PRODUCT!!!!
Test case 22: passed
=====
DOT PRODUCT!!!!
Test case 23: passed
=====
DOT PRODUCT!!!!
Test case 24: passed
=====
DOT PRODUCT!!!!
Test case 25: passed
=====
DOT PRODUCT!!!!
Test case 26: passed
=====
DOT PRODUCT!!!!
Test case 27: passed
=====
DOT PRODUCT!!!!
Test case 28: passed
=====
DOT PRODUCT!!!!
```

A small 'Snipping' tool window is visible on the right side of the IDE.

C: > Users > Jacob Sunia > Desktop > Vector.py

Project ... Vector.py cases.py

Run: cases

```
Test case 17: passed
=====
VECTOR SUBTRACTION!!!!
Test case 18: passed
=====
VECTOR SUBTRACTION!!!!
Test case 19: passed
=====
VECTOR SUBTRACTION!!!!
Test case 20: passed
=====
DOT PRODUCT!!!!
Test case 21: passed
=====
DOT PRODUCT!!!!
Test case 22: passed
=====
DOT PRODUCT!!!!
Test case 23: passed
=====
DOT PRODUCT!!!!
Test case 24: passed
=====
DOT PRODUCT!!!!
Test case 25: passed
=====
DOT PRODUCT!!!!
Test case 26: passed
=====
DOT PRODUCT!!!!
Test case 27: passed
=====
DOT PRODUCT!!!!
Test case 28: passed
=====
DOT PRODUCT!!!!
Test case 29: passed
=====
DOT PRODUCT!!!!
Test case 30: passed
=====

Process finished with exit code 0
|
```